Réseaux 5
La couche réseau
plan de contrôle

Juliusz Chroboczek

5 octobre 2020

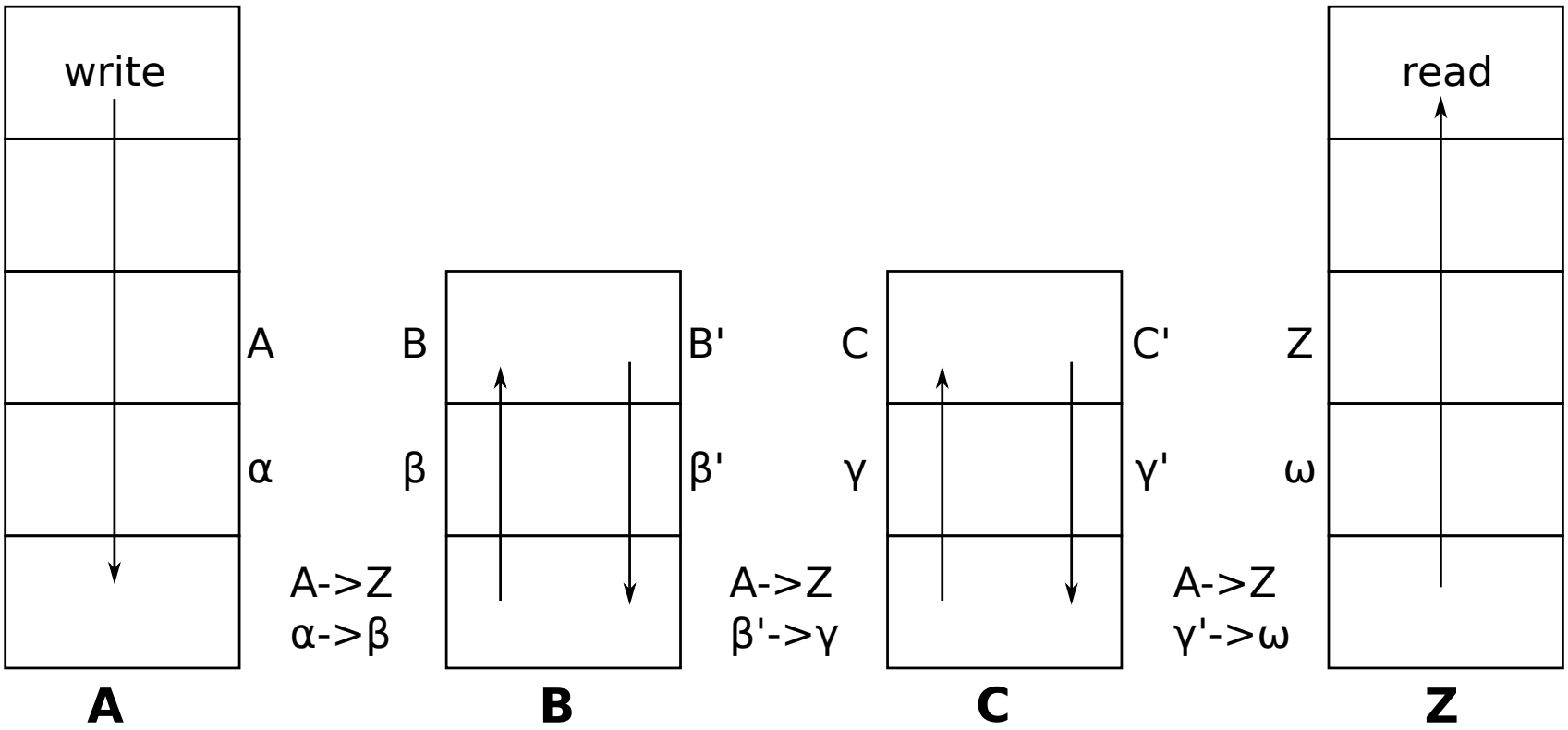
Plan de contrôle

La couche réseau est divisée en deux *plans* :

- le plan de données s'occuppe de pousser les paquets
  plusieurs millions de fois par seconde ;
- le plan de contrôle calcule par où il faut les pousser
  latence de l'ordre de la seconde.

La semaine dernière $\longrightarrow$ plan de données.
Cette semaine $\longrightarrow$ plan de contrôle.

Rappel



Avant de transférer un paquet, un routeur consulte la table de routage afin de déterminer l'adresse IP du voisin.
Le plan de contrôle construit la table de routage.

Tables de routage

Conceptuellement :

$$\mathrm{IP} \mapsto (\mathrm{if}, \mathrm{nh})$$

Implémenté naïvement,

- 4 milliards d'entrées en IPv4 ;
- ∞ entrées en IPv6.

La table de routage est compressée en utilisant des préfixes :

$\mathrm{P} \mapsto (\mathrm{if}, \mathrm{nh})$	(entrée normale)
$\mathrm{P} \mapsto \mathrm{if}$	(entrée connectée)

Tables de routage (2)

Entrée normale :

$$P \mapsto (\text{if}, \text{nh})$$

signifie

$$a_1 \mapsto (\text{if}, \text{nh})$$
$$a_2 \mapsto (\text{if}, \text{nh})$$

Entrée connectée :

$$P \mapsto \text{if}$$

signifie

$$a_1 \mapsto (\text{if}, a_1)$$
$$a_2 \mapsto (\text{if}, a_2)$$

Règle du préfixe le plus long

Les préfixes ne sont pas forcément disjoints — la table de routage peut être ambigüe.

Propriété : deux préfixes sont soit disjoints, soit l'un est inclus dans l'autre.

En cas d'ambigüité, c'est l'entrée la plus spécifique qui s'applique. C'est la règle du préfixe le plus long.

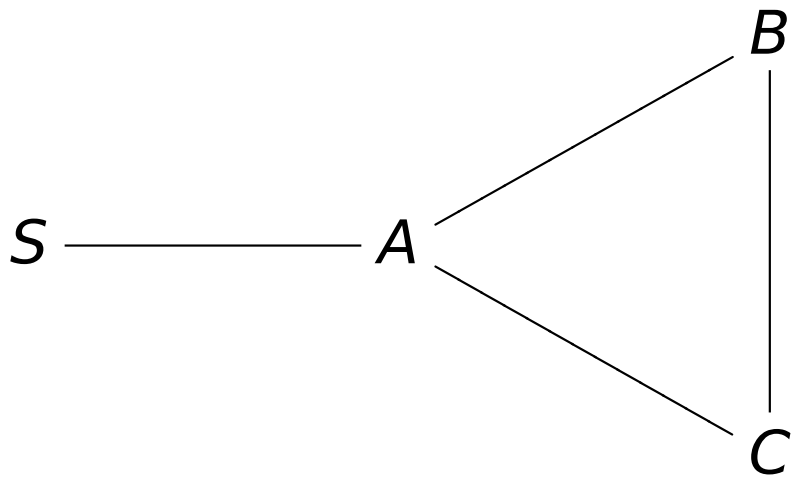
Plan de contrôle et plan de données

La couche réseau est modulaire :

- le plan de données ou *forwarding* s'exécute pour chaque paquet :
  - 10^7 fois par seconde (!) ;
  - consulte la table de routage.
- le plan de contrôle ou protocole de routage construit la table de routage :
  - latence permissible : 10 ms à 10 s (tranquille).

La seule interaction est à travers la table de routage.

Protocole d’accessibilité naïf



S	(S,S)	(S,S),(A,A)	(S,S),(A,A)	(S,S),(A,A),(B,A)
A	(A,A)	(A,A)	(S,S),(A,A),(B,B)	(S,S),(A,A),(B,B)
B	(B,B)	(A,A),(B,B)	(A,A),(B,B)	(S,A),(A,A),(B,B)
C	(C,C)	(A,A),(C,C)	(A,A),(B,B),(C,C)	(S,A),(A,A),(B,B)

Protocole d'accessibilité naïf

Notation : $\mathrm{nh}_X(Y) = Z$ signifie que Y route vers X à travers Z.

Initialement :

- la table de routage de Y ne contient que l'entrée $\mathrm{nh}_Y(Y) = Y$.

Périodiquement,

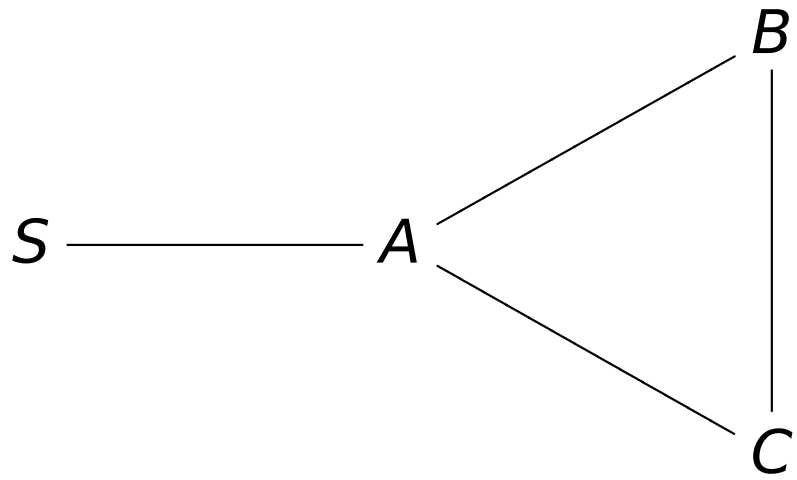
- Y envoie à tous ses voisin une *annonce* qui contient l'ensemble $\mathrm{dom}(R(X))$, c'est-à-dire l'ensemble des nœuds que X sait joindre.

Lorsqu'un routeur X reçoit une annonce $R(Y)$ de la part d'un voisin Y, pour chaque $S \in R(Y)$,

- si $S \notin R(X)$, alors $\mathrm{nh}_S(X) := Y$;
- sinon rien.

Protocole d'accessibilité naïf

Deuxième formulation



On fixe une destination S, on ne s'interesse qu'aux routes vers S.

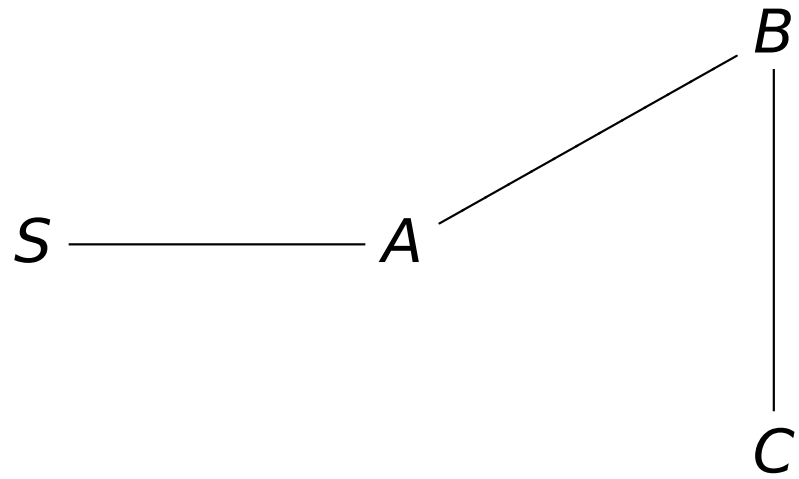
S	nh = S	nh = S	nh = S	nh = S
A	nh = $\perp$	nh = S	nh = S	nh = S
B	nh = $\perp$	nh = $\perp$	nh = A	nh = A
C	nh = $\perp$	nh = $\perp$	nh = A	nh = A

Protocole d'accessibilité naïf

Deuxième formulation

- Initialement $\mathrm{nh}(S) = S$, $\mathrm{nh}(X) = \bot$.
- Assez souvent, si $\mathrm{nh}(Y) \neq \bot$ alors Y envoie une annonce pour S.
- Lorsque X reçoit une annonce pour S de la part d'Y, alors :
  - si $\mathrm{nh}(X) = \bot$, alors $\mathrm{nh}(X) := Y$;
  - sinon rien.

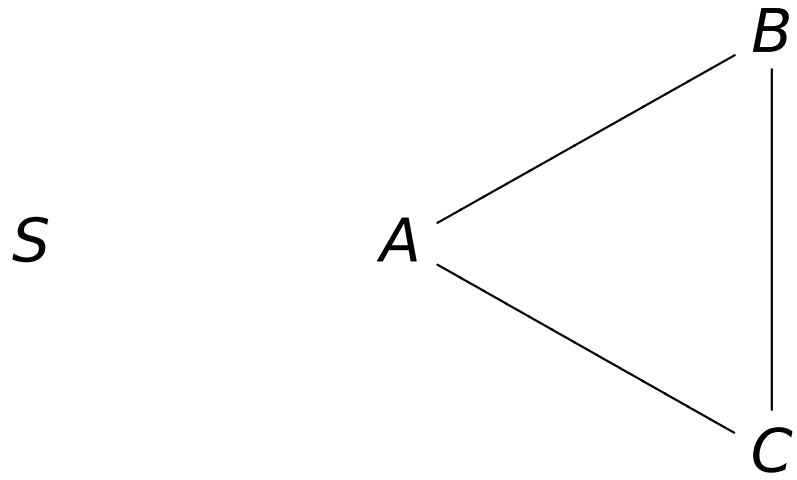
Timeout



– Si X n'entend plus Y pendant un temps t_{hold}, alors $\text{nh}(S) := \bot$.

S	nh = S	nh = S	nh = S
A	nh = S	nh = S	nh = S
B	nh = A	nh = A	nh = A
C	nh = A	nh = $\bot$	nh = B

Boucle de routage persistante



S	nh = S	nh = S	nh = S
A	nh = S	nh = ∞	nh = B
B	nh = A	nh = A	nh = A
C	nh = A	nh = $\perp$	nh = B

Métriques

On affecte à chaque lien un *coût*, un élément de $\mathbf{N}_+ \cup \{\infty\}$.

La *métrique* d'une route est la somme des coûts des liens qui la constituent.

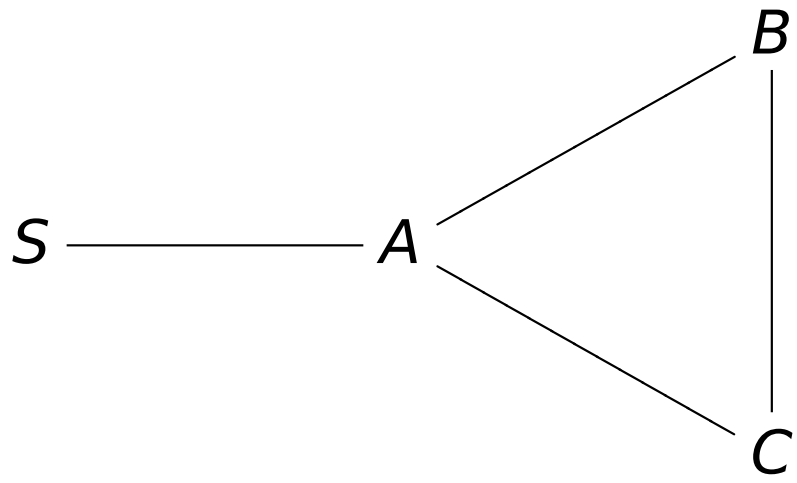
L'algorithme de routage calculera les chemins de métrique minimale.

Exemples :

- nombre de sauts : $c = 1$;
- délai ;
- 1/debit ;
- ...

L'algorithme s'en fiche (mais attention à la stabilité).

Protocole à vecteur de distances



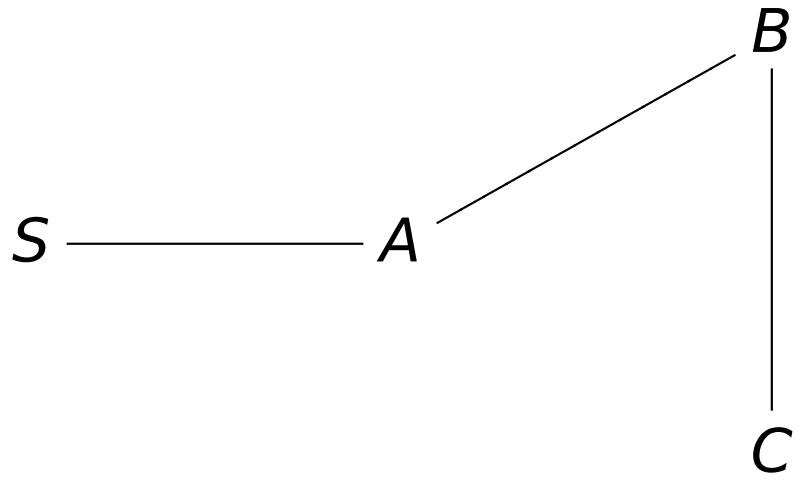
S	0	0	0	0
A	∞	1, nh = *S*	1, nh = *S*	1, nh = *S*
B	∞	∞	2, nh = *A*	2, nh = *A*
C	∞	∞	2, nh = *A*	2, nh = *A*

Convergence en $O(\Delta)$ — on ne peut pas faire mieux.

Protocole à vecteur de distances

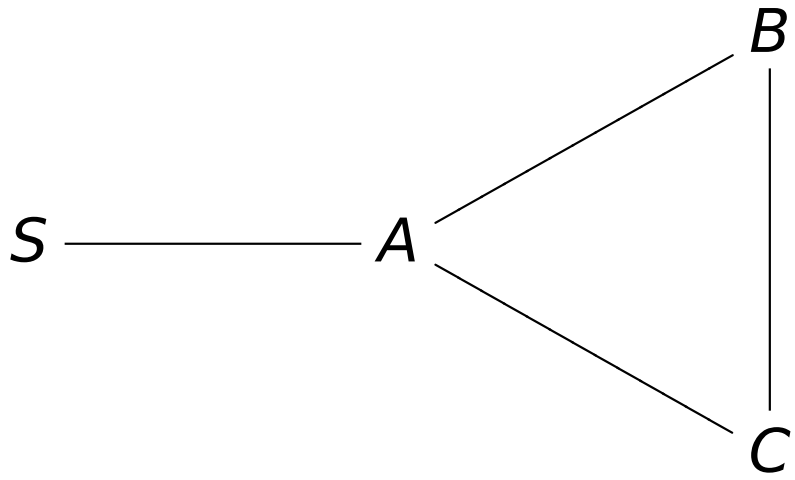
- Initialement, $d(S) = 0$ et $d(X) = \infty$.
- Assez souvent, si $d(Y) < \infty$, Y annonce $d(Y)$ à ses voisins.
- Quand X reçoit $d(Y)$:
  - si $nh(X) = Y$, alors $d(X) := c_{XY} + d(Y)$;
  - si $c_{XY} + d(Y) < d(X)$, alors $d(X) := c_{XY} + d(Y)$ et $nh(X) := Y$.
- *Timeout* : si $nh(X) = Y$, et X n'entend plus Y , $d(X) := \infty$ et $nh(X) := \perp$.

Protocole à vecteur de distances



S	0	0	0
A	1, nh = *S*	1, nh = *S*	1, nh = *S*
B	2, nh = *A*	2, nh = *A*	2, nh = *A*
C	2, nh = *A*	∞	3, nh = *B*

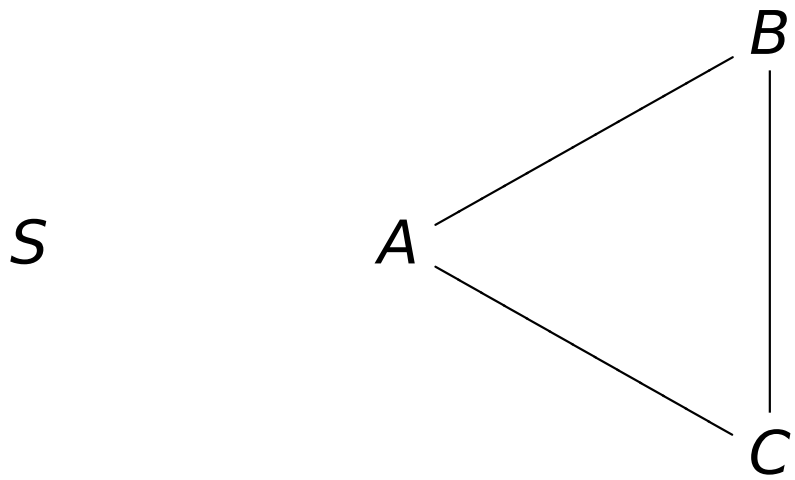
Protocole à vecteur de distances



S	0	0
A	1, nh = *S*	1, nh = *S*
B	2, nh = *A*	2, nh = *A*
C	3, nh = *B*	2, nh = *A*

Protocole à vecteur de distances

Comptage à l'infini



S	0	0	0	0	0	
A	1, nh = *S*	∞	3, nh = *B*	3, nh = *B*	3, nh = *B*	3,
B	2, nh = *A*	2, nh = *A*	2, nh = *A*	3, nh = *C*	3, nh = *C*	3,
C	2, nh = *A*	2, nh = *A*	2, nh = *A*	2, nh = *A*	∞	4,

Convergence en $O(\infty)$ —
il faut choisir de petites valeurs pour l'infini.

Métriques bornées

Depuis l'état initial, le vecteur de distances converge en $O(\Delta)$ — on ne peut pas faire mieux.

Après un changement de topologie, convergence en $O(\infty)$.

Les métriques ne sont pas dans **N**, mais dans un espace borné :

$$\{1, 2, \ldots n - 1, \infty\}.$$

Pour avoir une convergence rapide, il faut choisir une petite valeur pour ∞. (RIP : ∞ = 16.)

Conséquences :

- réseau de taille limitée ;
- limite la flexibilité des métriques.

Comptage à l'infini : réaction

Comptage à l'infini : grosse déprime :

- années 1980 : développement d'une nouvelle catégorie d'algorithmes de routage : état de lien ;
- années 1990-2000 : algorithmes à vecteurs de distance sans boucles :
  - vecteur de chemins (BGP) ;
  - vecteur de distances avec faisabilité.

Plan :

- vecteur à distances sans boucles ;
- état de lien.

Digression : routage hiérarchique

Le routage dans l'Internet est hiérarchique :

- l'Internet est divisé en systèmes autonomes (AS)
  - le campus est un (petit) AS ;
  - *Orange* est un (moyen) AS ;
- entre AS : routage externe ;
- à l'intérieur de chaque AS : routage interne.

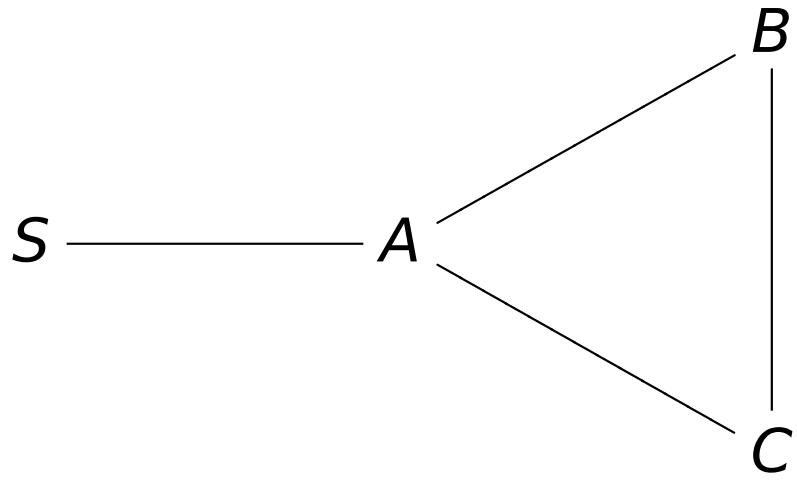
Hors d'un AS, la topologie interne n'est pas visible :

- un changement de topologie interne (plantage d'un routeur) n'est pas propagé hors de l'AS ;
- le routage entre AS n'est pas forcément optimal.

Il y a un seul protocole de routage externe : BGP.
BGP est basé sur l'algorithme à vecteur de distances.

Chaque AS choisit son protocole de routage interne.

Vecteur de chemins



S	S	S	S
A		$A \cdot S$	$A \cdot S$
B			$B \cdot A \cdot S$
C			$C \cdot A \cdot S$

Vecteur de chemins

- Initialement, $\text{path}(S) = \epsilon$ et $\text{path}(X) = \bot$.
- Assez souvent, si $\text{path}(Y) \neq \bot$, Y annonce $Y \cdot \text{path}(Y)$ à ses voisins.
- Quand X reçoit un chemin p de Y :
  - si $\text{nh}(X) = Y$, alors $\text{path}(X) := p$;
  - si $|p| < |\text{path}(X)|$, alors $\text{path}(X) := p$.
- *Timeout* : si $\text{path}(X) = Y \cdots$, et X n'entend plus Y, alors $\text{path}(X) := \bot$.

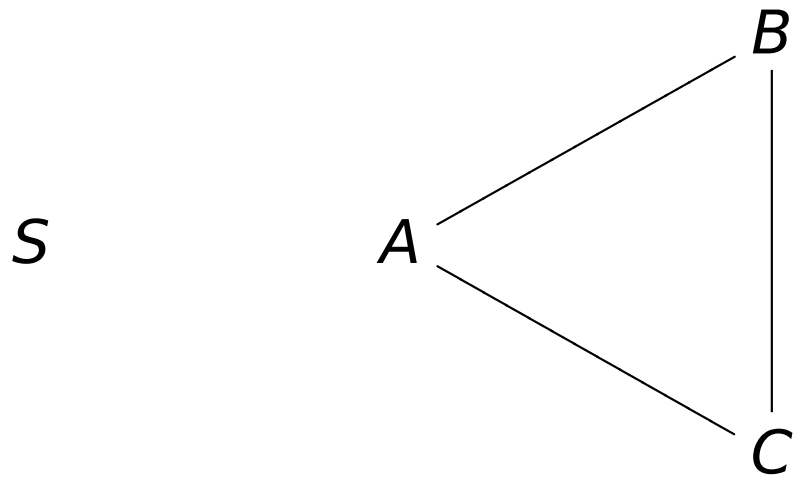
Avantages :

- évite les boucles persistantes (mais pas transitoires) ;
- permet un filtrage précis des routes.

Désavantages :

- les annonces sont de taille importante ;
- il faut annoncer les changements de chemin même si le *next hop* ne change pas.

Vecteur de distances avec faisabilité



S	nh = S	nh = S	nh = S
A	nh = S	nh = ∞	nh = B
B	nh = A	nh = A	nh = A
C	nh = A	nh = $\perp$	nh = B

On s'attendrait à avoir la propriété

$$X = \mathrm{nh}(Y) \longrightarrow \mathrm{d}(X) < \mathrm{d}(Y)$$

Peut-on trouver une notion plus fine que la métrique qui satisfasse cette propriété ?

Vecteur de distances avec faisabilité

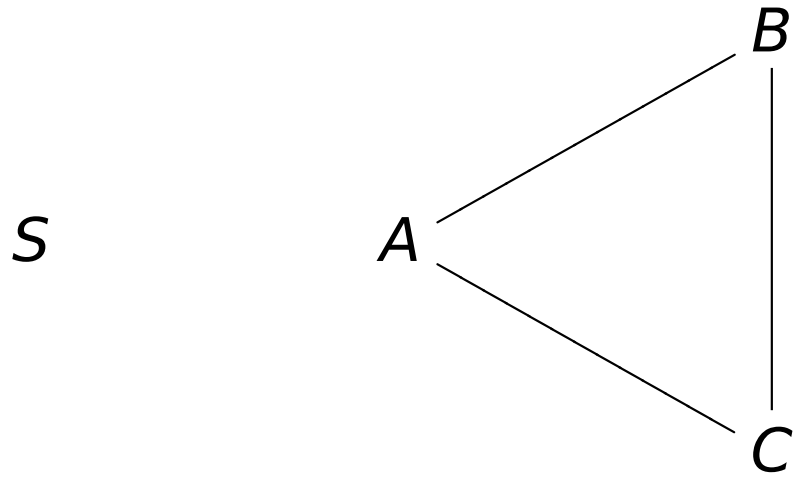
Chaque routeur maintient une distance de faisabilité :

$$\mathrm{fd}(X) = \min_{t \leq \mathrm{now}} \mathrm{d}(X, t)$$

Lorsque X reçoit une annonce,

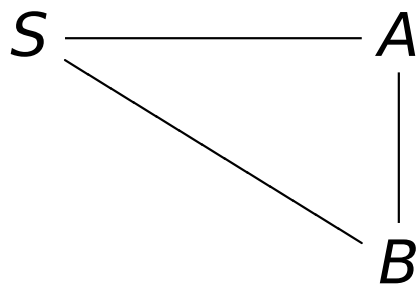
- si $d < \mathrm{fd}(X)$, l'annonce est acceptée ;
- sinon, l'annonce est ignorée (« infaisable »).

Vecteur de distance séquencé : exemple



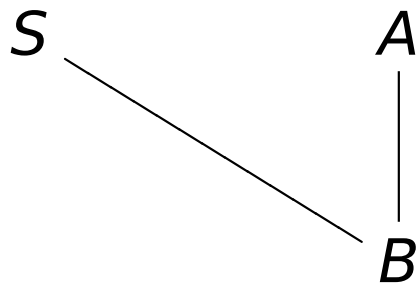
A	1, fd = 1	∞, fd = 1	∞, fd = 1	∞, fd = 1
B	2, fd = 2	2, fd = 2	∞, fd = 2	∞, fd = 2
C	2, fd = 2	2, fd = 2	∞, fd = 2	∞, fd = 2

Vecteur de distance séquencé : famine



$d(A) = 1, fd(A) = 1$

$d(B) = 1, fd(B) = 1$



$fd(A) = 1$

$d(B) = 1$

La seule route disponible est infaisable. Famine.

Besoin d'un mécanisme de résolution de famine :

- synchronisation globale (EIGRP) ;
- routes séquencées (DSDV, Babel).

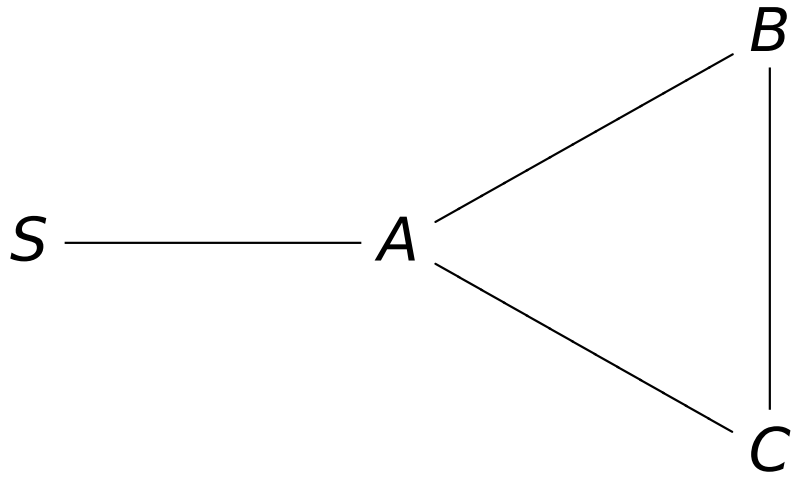
Protocoles à états de lien

La grosse déprime du début des années 1980 a mené au développement des algorithmes à états de lien.

Principe :

1. calcul de la topologie locale ;
2. inondation fiable ;
3. calcul de l'arbre des plus courts chemins et de la table de routage.

Calcul de la topologie locale



$$L(S) = \{(S, A)\}$$
$$L(A) = \{(A, S), (A, B), (A, C)\}$$
$$L(B) = \{(B, A), (B, C)\}$$
$$L(C) = \{(C, A), (C, B)\}$$

Inondation fiable

La topologie locale de chaque nœud est inondée de manière fiable.

Quand l'inondation a terminé, chaque nœud connaît la topologie globale.

Calcul de l'arbre des plus courts chemins

Après l'inondation globale, chaque nœud connaît la topologie globale.

Il peut donc calculer l'arbre des plus courts chemins depuis lui :

- Dijkstra ($O(n^2 \log n)$ si bien implémenté) ;
- Bellman-Ford ($O(n^3)$).

$$\mathrm{P}(S) = \{S, S \cdot A, S \cdot A \cdot B, S \cdot A \cdot C\}$$

Cet arbre est ensuite tronqué pour obtenir la table de routage :

$$\mathrm{nh}_S(S) = S \qquad \mathrm{nh}_B(S) = A$$
$$\mathrm{nh}_A(S) = A \qquad \mathrm{nh}_C(S) = A$$

Fragilité de l'état de lien

La correction de l'état de lien dépend de 3 hypothèses :

1. les bases de données de liens sont synchronisées ;
2. l'ensemble des plus courts chemins est un arbre ;
3. le recollage des arbres tronqués produit des routes sans boucles.

2 et 3 dépendent de la distributivité de l'algèbre de routage, vérifiée si les métriques sont des entiers.

1 est impossible à garantir. Probable si :

- implémentation soigneuse de l'inondation fiable ;
- domaine de routage de taille modérée.

Aires et pseudo-nœuds

Le routage à états de liens demande des domaines de routage de taille modérée.

Deux optimisations :

- le domaine de routage est découpé en aires,
  - état de lien à l'intérieur d'une aire ;
  - vecteur de distances entre les aires.
- les réseaux à accès multiples sont replacés par des pseudo-nœuds,
  - remplace $n(n-1)/2$ arêtes par n arêtes.

Conclusion

Deux grandes familles de protocoles de routage :

- vecteur de distances (DV) ;
- état de liens (LS).

Les deux marchent mal en version naïve. Pour que ça marche :

- DV $\longrightarrow$ vecteur de chemins ;
- DV + faisabilité + résolution de famine ;
- LS + aires + pseudo-nœuds.

Dans l'Internet de 2020 :

- vecteur de chemins dominant en externe ;
- état de liens dominant en interne.